

Санкт-Петербургский Политехнический Университет
Высшая школа прикладной математики и вычислительной физики, ФизМех
01.03.02 Прикладная математика и информатика

Отчет по лабораторной работе № 14
"Решение краевой задачи для ОДУ 2-ого порядка"
дисциплина "Численные методы"

Выполнил студент гр. 5030102/20003
Преподаватель

Ляпустин Е.О.
Козлов К.Н.

Июнь, 2024

Формулировка задачи и ее формализация

Формализация задачи:

Найти решение краевой задачи ОДУ второго порядка модифицированным методом суперпозиции (+метод Эйлера). Также найти решение заданой точности.

Поставленные задачи:

1. Иллюстрация работы метода. Построить графики точного и численных решений для двух фиксированных значений шага на отрезке. Построить график ошибки на отрезке для этих решений.
2. Исследование точности метода. Заданная точность достигается по правилу для метода на каждом шаге, построить график изменения шага по отрезку. Построить график зависимости фактической погрешности от заданной точности.

Алгоритм метода

Краевая задача:

$$p(x)y + q(x)y + r(x)y = f(x),$$

$$x[a, b]$$

$$\alpha_0 y(a) + \alpha_1 y'(a) = A$$

$$\beta_0 y(b) + \beta_1 y'(b) = B$$

ГУ:

$$\alpha_1 = 0, \beta_0 \neq 0, \beta_1 \neq 0$$

Поиск решения в виде $y(x) = u(x) + cv(x)$, где

$$L(u) = f$$

$$L(v) = 0$$

1. Построение двух задач Коши: из ГУ получаем начальные условия:
$$\begin{cases} u(a) = \frac{\alpha_0}{\alpha_0^2 + \alpha_1^2} A \\ u'(a) = \frac{\alpha_1}{\alpha_0^2 + \alpha_1^2} A \end{cases}$$

и
$$\begin{cases} v(a) = \alpha_1 \\ v'(a) = -\alpha_0 \end{cases}$$

2. Построение сеток u^h, v^h (использовал метод Эйлера)

3. Вычисление константы C:
$$C = \frac{B - \beta_0 u(b) - \beta_1 u'(b)}{\beta_0 v(b) + \beta_1 v'(b)}$$

Для достижения заданной точности использую правило Рунге:

1. Для начала шаг равен половине длины отрезка
2. Если $|y_{2i+2}(\frac{h}{2}) - y_{i+1}(h)| > \epsilon$ то шаг уменьшается вдвое.
3. Иначе: перехожу к следующему элементу сетки ($i+=1$)

Предварительный анализ задачи

Краевая задача:

$$y'' + \cos(x)y' + \sin(x)y = 1 - \sin(x) \quad (1)$$
$$[a, b] = [0, \pi/2]$$

Точное решение:

$$y = \sin(x)$$

Пусть

$$\alpha_0 = 1, \alpha_1 = 0, \beta_0 = 1, \beta_1 = 1$$

Тогда

$$A = 0, B = 1$$

Ручной расчет

Ручной расчет добавлен в репозиторий отдельным файлом (pp.pdf)

Модульная структура программы и контрольные тесты

Модульная структура программы

```
typedef struct {  
    double y1;  
    double y2;  
}vector;
```

- структура вектора $Y = (y(x), y'(x))^T$

```
double f(double x);  
double p(double x);  
double q(double x);
```

- функции соответствующие краевой задаче, (1)

```
double F(double x, vector y);  
double F_0(double x, vector y);
```

- явная запись уравнения для y'' (для уравнений с нулевым коэффициентом равным 0 и $f(x)$), (1)

```
vector F_vect(double x, vector y);  
vector F_vect_0(double x, vector y);
```

- правые части уравнений $Y' = F(x, Y)$, $Y = (y(x), y'(x))^T$

```
vector* find_initials(double A, double alpha_0, double alpha_1);
```

- функция, которая возвращает $u_0 = (u(a), u'(a))^T, v_0 = (v(a), v'(a))^T$ - начальные векторы

```
double* solve(vector* initials, double betta_0, double betta_1, double B,  
    double a, double b, vector(*F)(double, vector),  
    vector(*F_0)(double, vector), int n);
```

- функция, которая ищет решение краевой задачи для фиксированного шага (число n - число узлов сетки)

```
double** solve_runge(vector* initials, double betta_0,  
    double betta_1, double B, double a, double b,  
    vector(*F)(double, vector),  
    vector(*F_0)(double, vector), double eps, int* n_, int* len_h);
```

- функция, реализующая пошаговый контроль точности eps, возвращает двумерный массив, содержащий сетку x^h и y^h , массив выполненных шагов, массив всех шагов.

Численный анализ решения

Иллюстрация работы метода

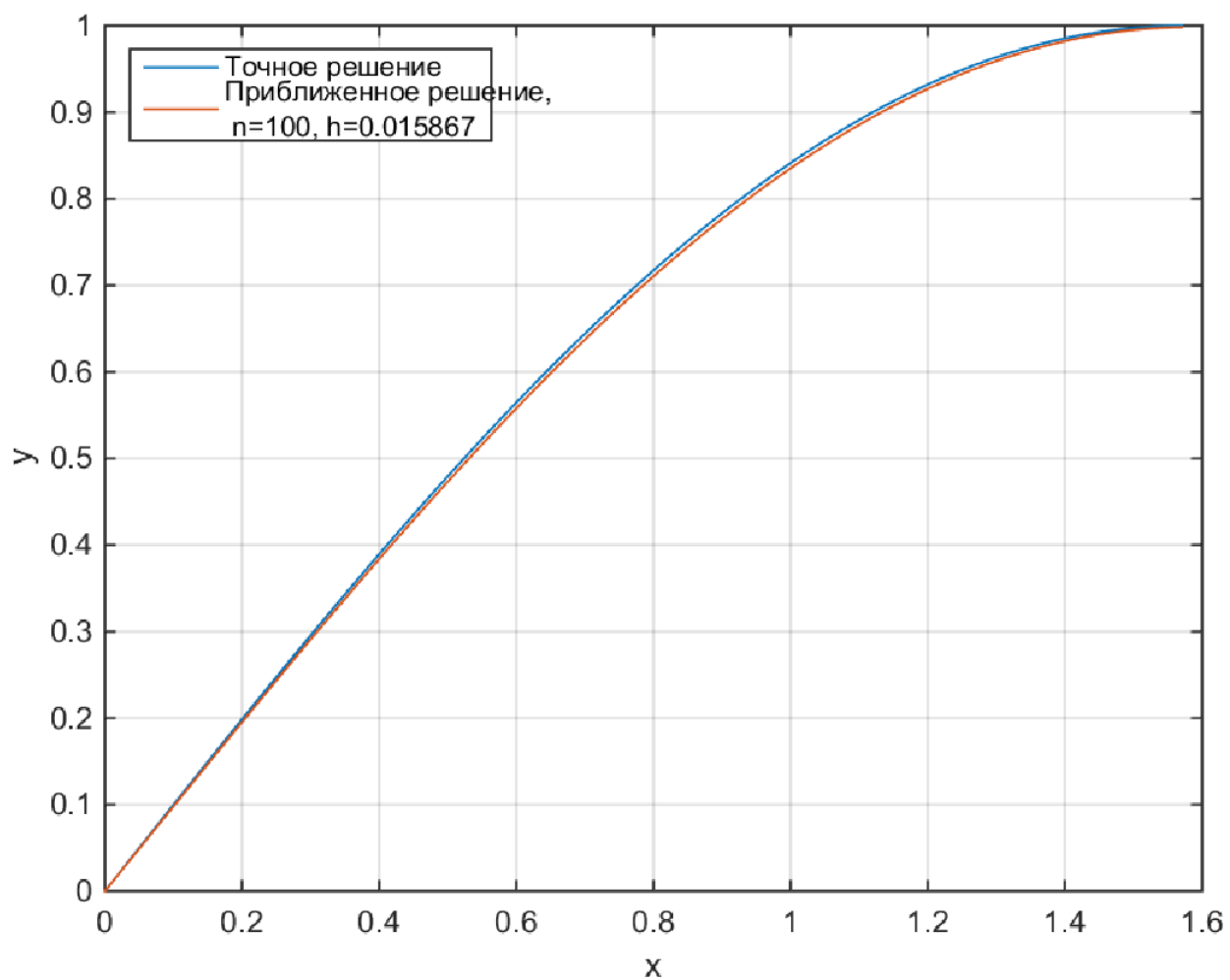


Рис. 1: Графики точного и приближенного решений ($n=100$)

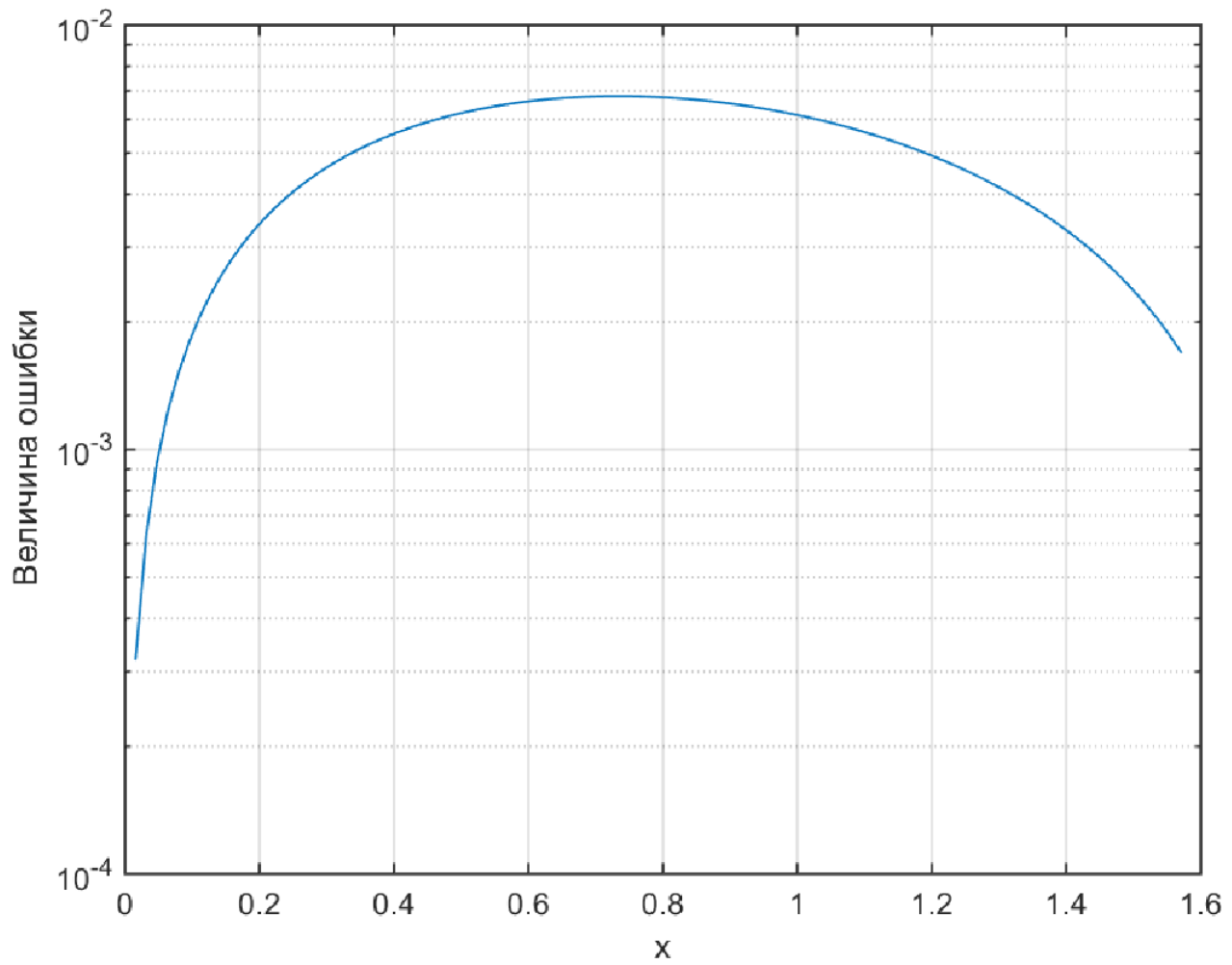


Рис. 2: График поточечной ошибки ($n=100$)

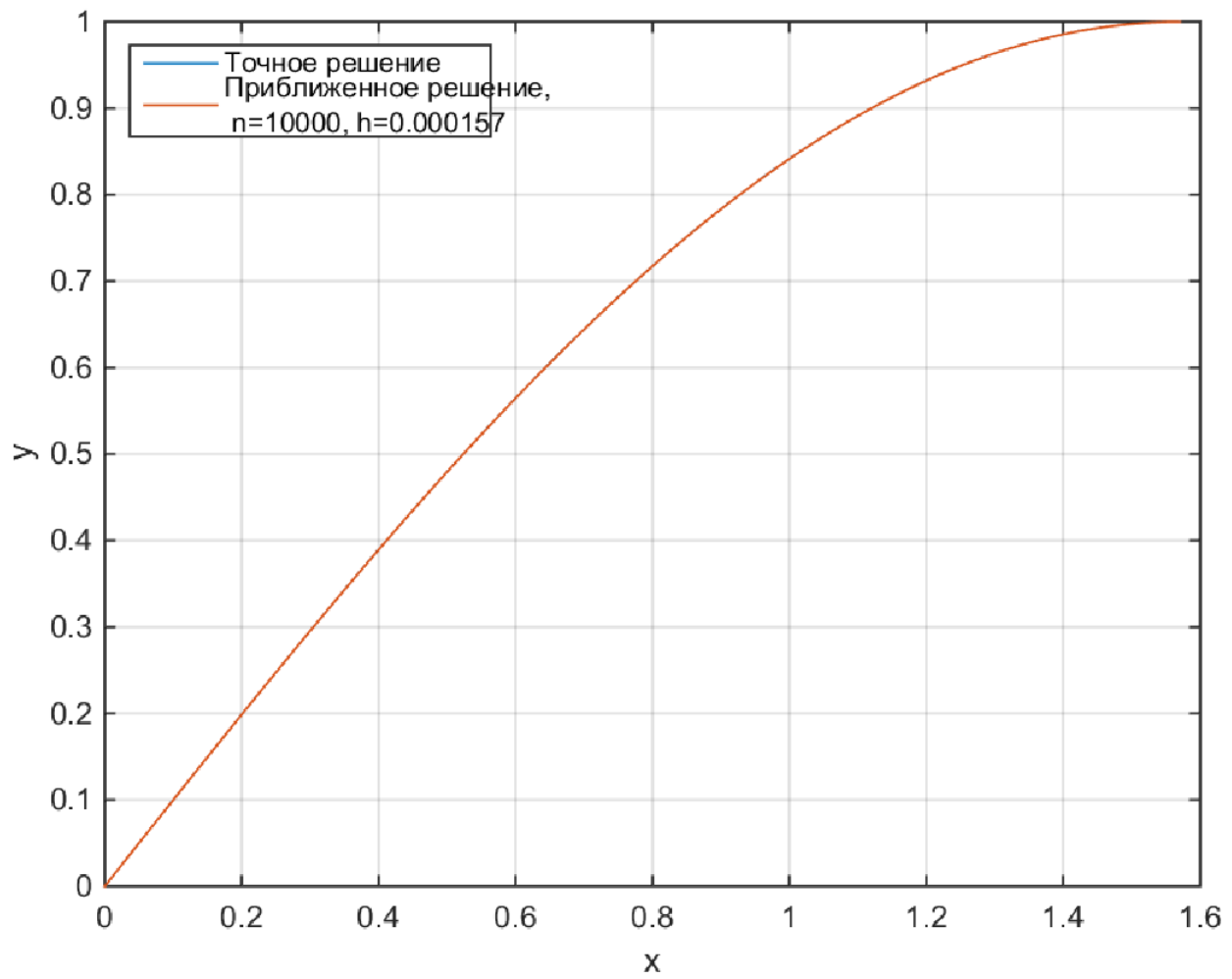


Рис. 3: Графики точного и приближенного решений ($n=10000$)

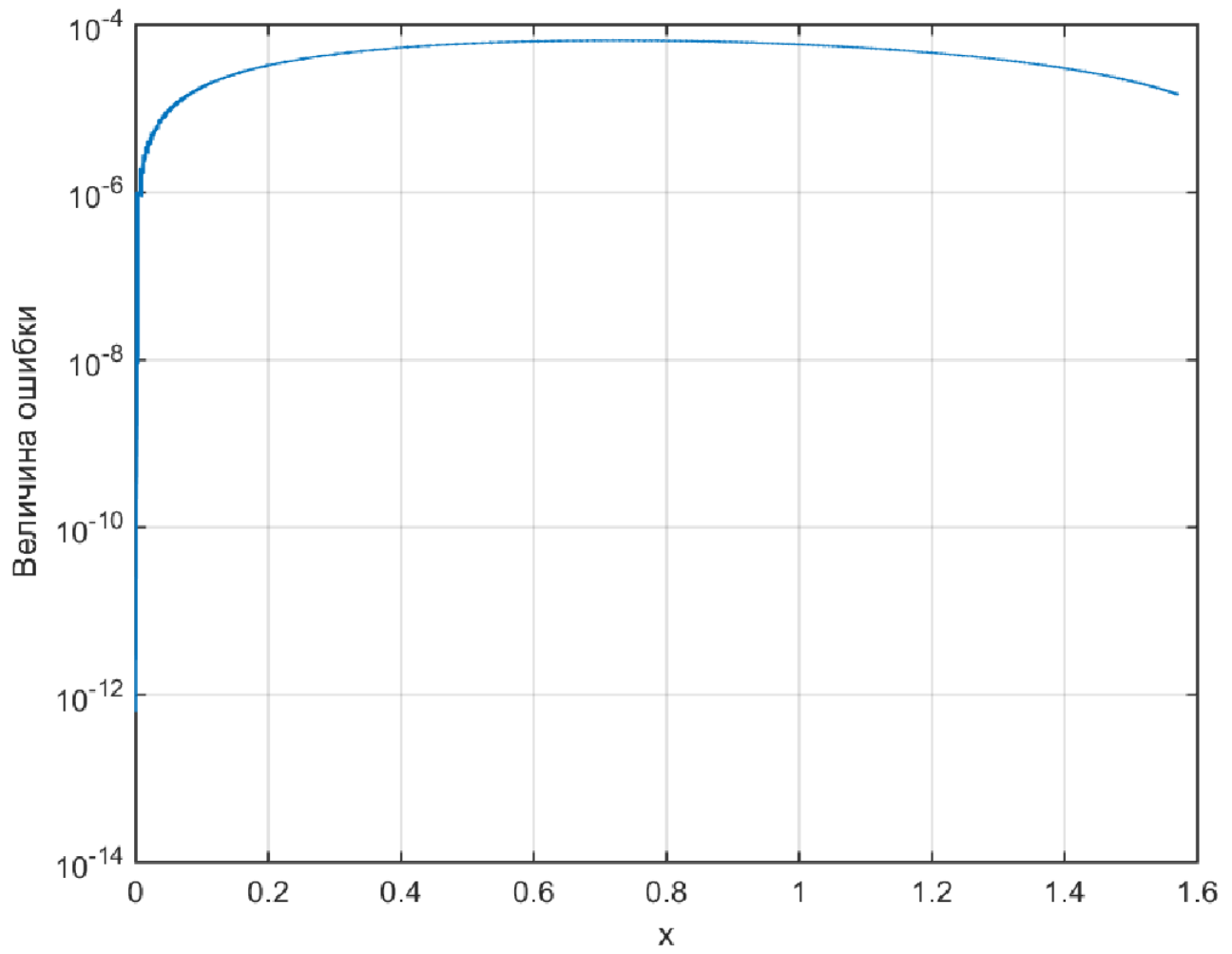


Рис. 4: График поточечной ошибки ($n=10000$)

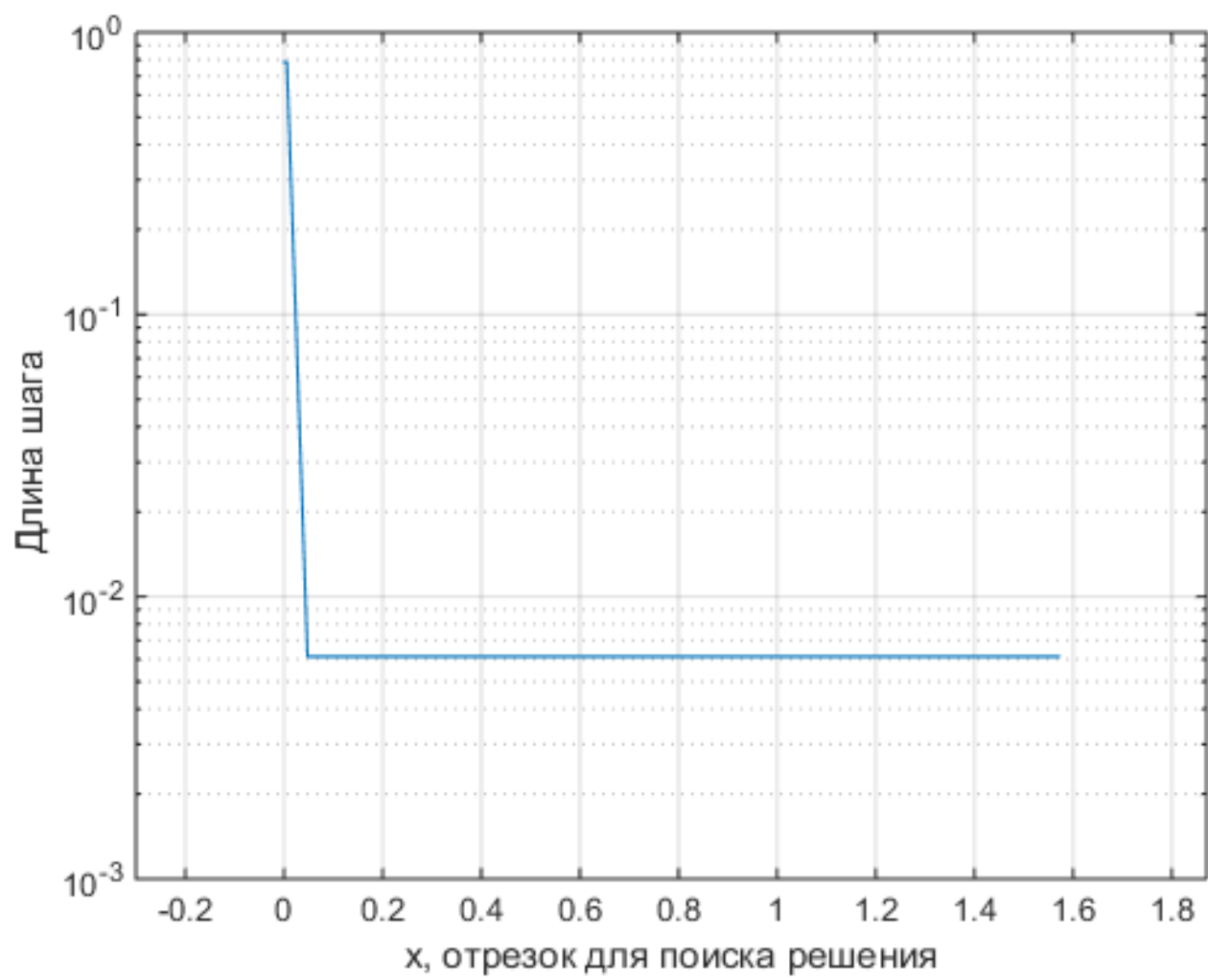


Рис. 5: График изменения шага по отрезку поиска решений при заданной точности 10^{-2}

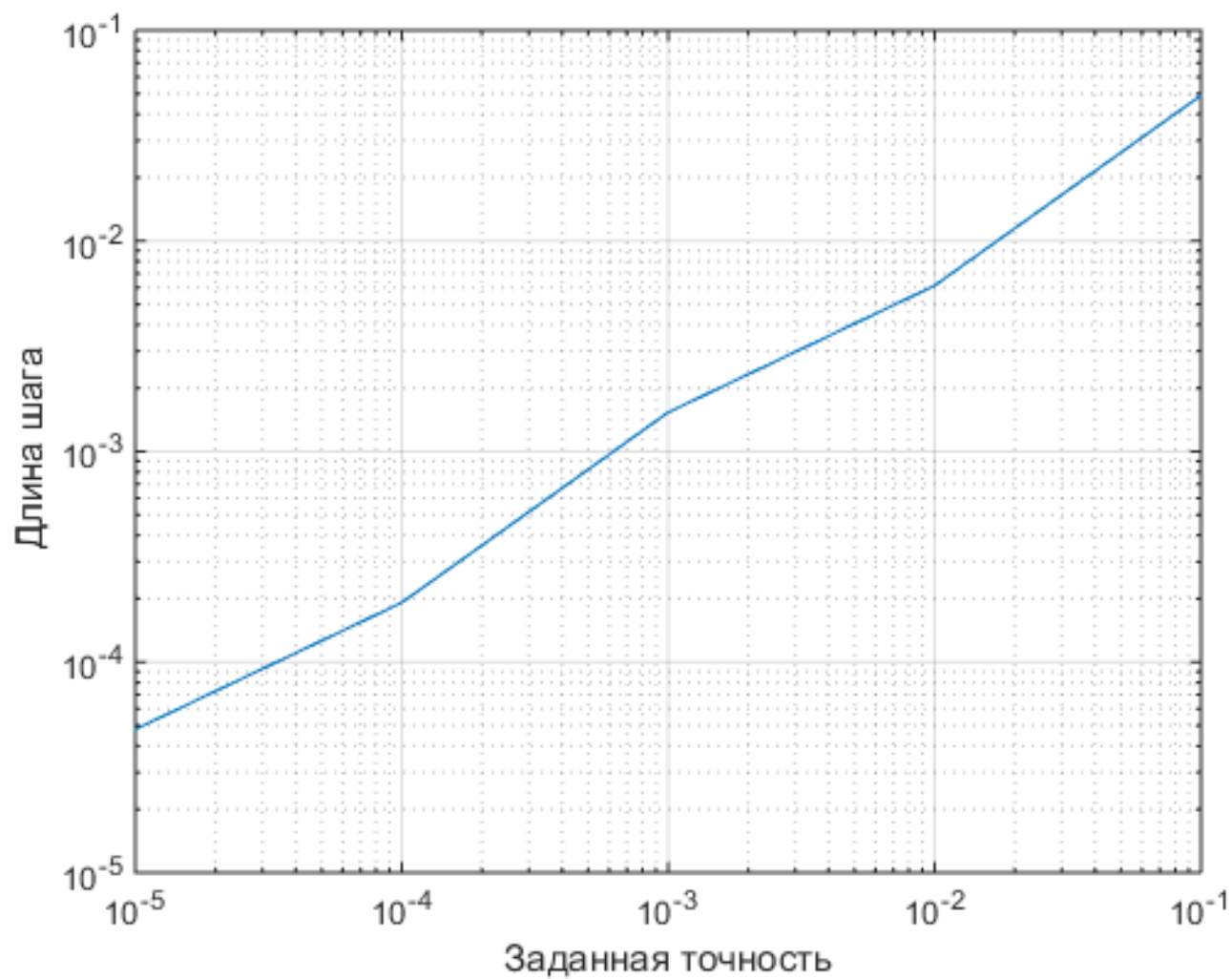


Рис. 6: График зависимости длины шага разбиения от заданой точности

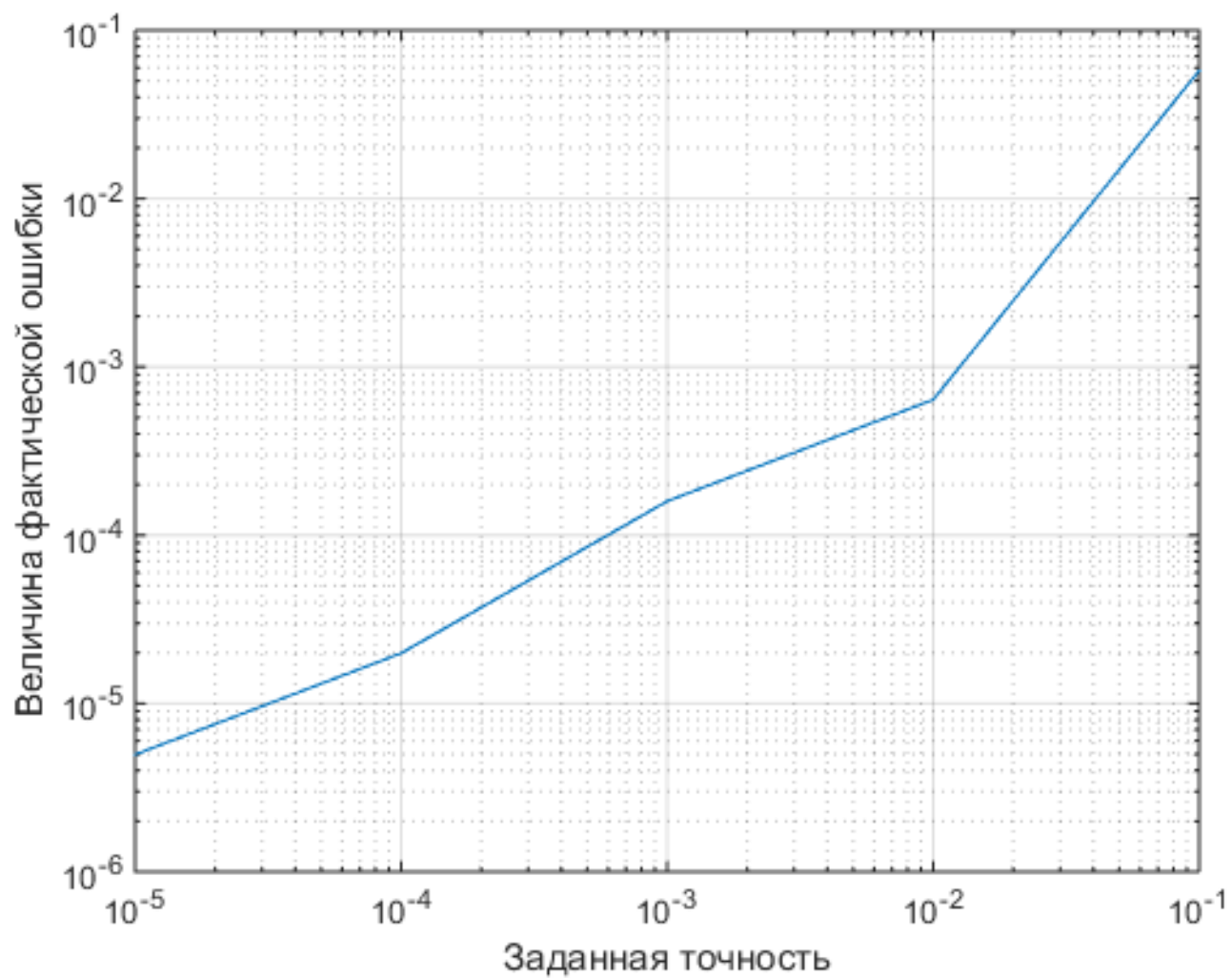


Рис. 7: График зависимости фактической ошибки от заданной точности

Выводы

1. Из рисунков (2) и (4) видно, что с уменьшением шага в 100 раз максимальная ошибка уменьшается в 100 раз (что соответствует методу Эйлера).
2. Из рис. (7) видно, что заданная точность достигается